

A small laboratory automation system has been developed by using the IBM 5100 Portable Computer in conjunction with the Research Device Coupler. This compact system provides a dedicated, high-level-language computer and a versatile data acquisition and control interface for experiments in which data rates do not exceed 9600 baud. Two experiments exemplify the use of the system.

The Research Device Coupler described in this paper is a prototype of the IBM 7406 Device Coupler.

The IBM 5100 and the Research Device Coupler— A personal laboratory automation system

by H. Cole and A. A. Guido

Scientists and engineers who use computers as part of their test instrumentation normally choose an event-driven computer, such as the IBM System/7, or they interface their equipment to a time sharing system, such as TSO.¹ In either case, it is necessary for the experimentalist to have acquired skill in programming. With time sharing systems, this problem is alleviated somewhat by the availability of high-level languages such as APL, BASIC, interactive FORTRAN, and PL/I. However, time sharing systems often are not responsive enough or available over long enough periods of time. These limitations become especially noticeable with experiments in which run time is extremely long or the nature of the experiment requires a dedicated facility (for example, when the sample is destroyed during testing, as in gas chromatography).

For many experiments in this class, requirements for data capture and instrument interfacing can be met using a small computer and a functional data acquisition and control mechanism. At IBM's Thomas J. Watson Research Center, we attached a portable computer, the IBM 5100, to the Research Device Coupler, an instrumentation interface, to create an easy-to-use laboratory automation system for the individual experimentalist. This paper describes two experiments which have been operated successfully with the 5100 and the Research Device Coupler.

The Research Device Coupler (RDC)^{8,9,10} is designed as an instrumentation interface for experimentalists using interactive, terminal oriented systems. It is a 19-inch-wide, rack mountable cage with positions for eight hardware modules (see Figure 2). Data and control lines are contained in a back plane, or interface highway, and the cage has self-contained power supplies.

The Research Device Coupler

In addition to the standard keyboard, character screen, and tape cassette unit, the 5100 has available as optional equipment a second tape cassette unit, a printer, a communications adapter, and a serial input/output (SIO) adapter. The communications adapter⁶ enables the 5100 to be used as a terminal in conjunction with large, interactive systems. The SIO adapter⁷ provides for the attachment and local control of devices with a communications feature. The Research Device Coupler has such a feature, so the two units can work together with no change in either.

The IBM 5100 Portable Computer (see Figure 1) provides the scientist or engineer with personal, local computing using an interactive high-level language, either APL^{2,3} or BASIC.^{4,5} With maximum memory in the machine, the user has an active workspace of approximately 60K bytes; and with one or two tape cassettes, an additional 200K or 400K bytes are at his disposal.

The IBM 5100

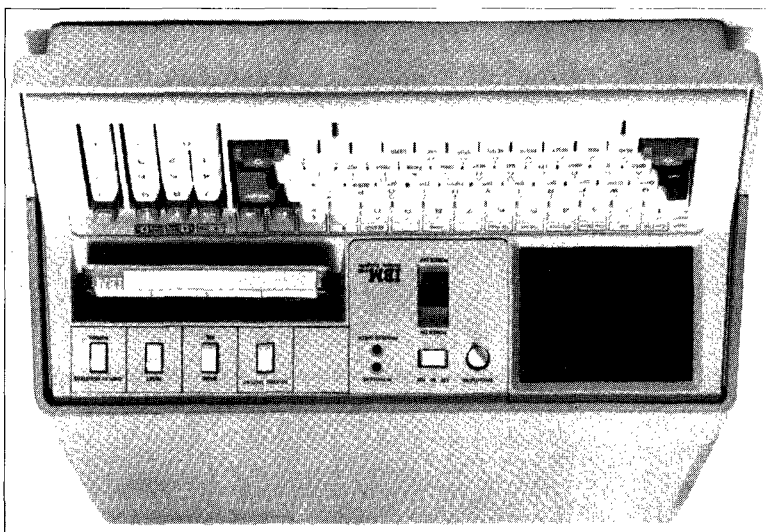
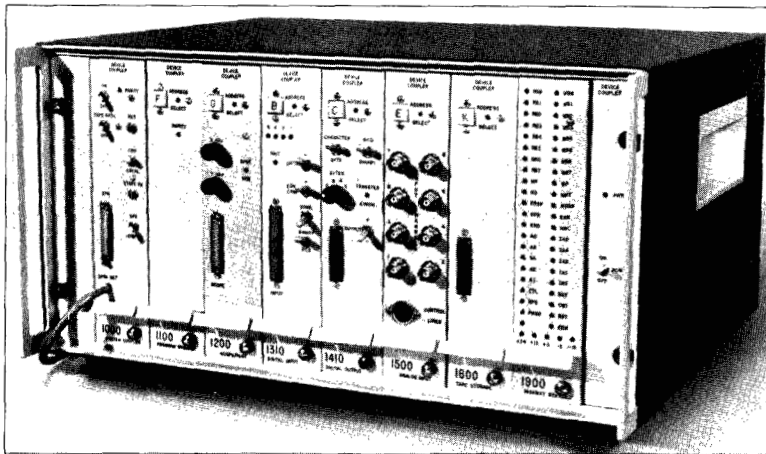


Figure 1 The IBM 5100 Portable Computer

Figure 2 The Research Device Coupler



The hardware modules provide for system communications, program storage, digital and analog input and output, and tape storage. Except for the system communication module and the program storage module (a loadable local sequencer), the modules can be used in any combination. Each module is independently addressable regardless of its position in the cage.

The system communication module permits the RDC to be interposed between the terminal and its modem, and, in essence, provides an input/output bus for the terminal. No hardware change is required in the terminal or modem. The digital input module permits the attachment and operation of digital devices such as multichannel analyzers and digital voltmeters. The digital output module can be used to turn lines on and off in order to control such devices as steppers, lights, switches, and motors. The analog input module contains an analog-to-digital converter for direct digitization of analog voltage signals. The analog output module produces two voltage outputs, used most often to drive graphic plotting devices. The tape storage module provides for local storage of data on a tape cassette. In addition, the RDC can include a highway status module, which has indicator lights for each line in the interface highway. Detailed module specifications are given in references 8, 9, and 10.

The RDC is activated by messages, which can be generated in any high-level language. The messages are made up of an operation code (any of five characters), a module address character, a function digit (if other than the default function is wanted), and whatever string of data characters the module may need. Data can range from none to a long character string, as for plotter

Table 1 Examples of parameters that can be passed to input/output devices by means of shared variables

<i>Device address</i>				
BASIC	APL	Use	Direction	Function
A08	31	Command	OUT	Passes output data across input/output interface
A04	32	DATA	OUT	Passes output data across input/output interface
A02	33	DATA	IN	Requests data input via input/output interface

Table 2 Partial list of 5100 command options

<i>Function format</i>	<i>Meaning</i>
I/N	Set input buffer size in bytes, $N \geq 3$
O/N	Set output buffer size in bytes, $N \geq 3$
R/N	Set input/output communications rate, $20 \leq N \leq 9600.5$ baud
A/A or B	Select input and output translation tables (A = APL, B = BASIC)
K/T, M, I, or S	Select mode; S is required for RDC operations

output. The five operation characters are: I (write), H (write with function digit to follow), R (read), \$ or, in APL, [(write transparent), and Q (start program storage module). Any of 16 characters can be used for module addressing, and the numerals 1 through 9 are used for module function selection.

As an example, assume that a digital input module has the address B, and that we wish to read it using module function 1 (that is, we wish to transfer into the memory of the 5100 the contents of some digital device cabled to the digital input module). The activating message would be RB1, which would instruct the RDC to read data from module B as directed by module function 1. No data are needed in this message. Function 1 causes the digital input module permits the attachment and operation of digital devices, then places itself in the WAIT state until the external device returns a ready signal, then transfers the digits via its input lines, through the system communication module, into the 5100. Up to 64 lines (16 hexadecimal characters) can be wired in parallel into the digital input module. The user can preset the number of digits to be returned to the 5100 for each measurement.

Strings of messages can be catenated to reduce the number of input/output operations. For example, the message IB50IC10RB1 instructs the system to write the number 50 (perhaps the number of readings to be taken) into module B, then write the number 10 into module C, and then read module B as above.

5100-RDC attachment

The 5100 deals with input/output devices through shared variables. The user defines the name of a variable and, using a system command, offers to share it with the 5100's auxiliary processor. Each time the user assigns the variable a new value, say a character string, the auxiliary processor passes the variable to the appropriate device; or if the variable name is used in a program, the system gets the next input from the specified device. Table 1 lists some of the operations that can be accomplished by means of shared variables. Table 2 lists some of the many options that can be selected and passed to the internal 5100 controller.

For example, to set up a device for an input of 130 characters (128 data characters plus beginning-of-text and end-of-text characters), at a 300-baud transmission rate, an APL user would open the device as follows:

```
.
.
.
[3]  A OFFER GLOBAL VARIABLE 'DC' TO SYSTEM
[4]  1 □SVO'DC'
[5]  A OPEN DEVICE 31 (COMMAND PROCESSOR)
[6]  DC ← 'OUT 31001 TYPE=I'
[7]  A SELECT COMMAND PROCESSOR OPTIONS
[8]  DC ← 'K/S,R/300,I/130'
[9]  A EXIT COMMAND PROCESSOR
[10] DC ← 10
.
.
.
```

Statement 8 above selects terminal option mode S and a rate of 300 baud, and it specifies an input buffer of 130 characters.

The characteristics of the SIO adapter on the 5100 and those of the system communication module on the RDC both conform to the RS-232C standards, so the two units are directly compatible. One need merely connect them using a standard cable supplied with the 5100.

Example applications

the force magnetometer

The magnetic properties of a material can be determined by measuring its magnetic moment, or susceptibility, by the Faraday method. The measurement is made using a force magnetometer, an instrument consisting of a microbalance that measures the pull on a sample in a high magnetic field (up to 20 kilo-oersteds) over a wide range of temperatures. The sample is inserted into a cryogenic chamber, and its temperature is reduced in steps to that of liquid helium (4 degrees Kelvin). At each temperature step, the magnetic field is swept from zero to some preset value by a local controller, and the force exerted on the sample is measured. The three parameters of interest—force, field, and temperature—are represented as analog signals. Normally these signals would be recorded on a strip chart.

The complete testing of a sample usually requires several hours with this experimental setup. The data rates are extremely slow, and the measurements themselves are time consuming. Typically, eight measurements are made at each temperature setting, one measurement every 20 seconds. Usually measurements at 30 temperature settings are required, each setting taking about 15 minutes to reach thermal equilibrium. In addition, it may take two hours or more to transcribe data from a strip chart manually and, using an APL terminal, enter it into a computer for analysis and plotting.

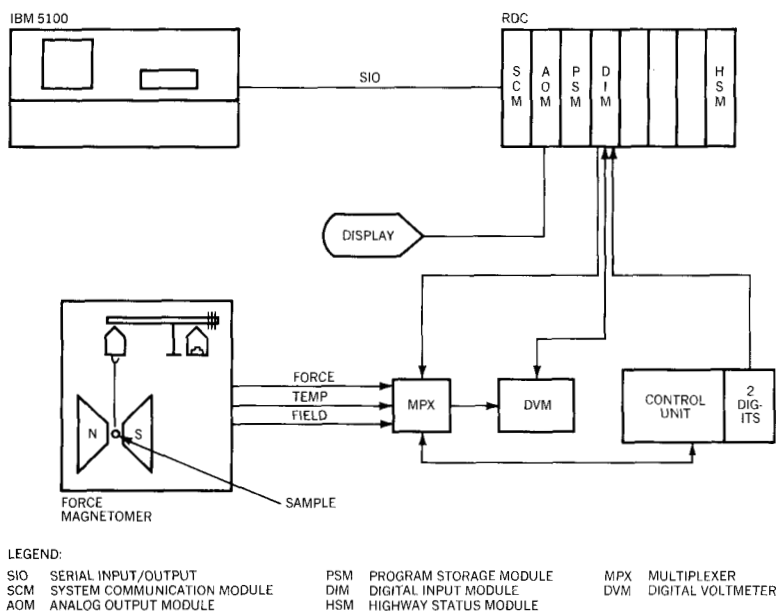
The manual transcription of data, a tedious and error-prone process, can be eliminated by multiplexing the analog signals into a digital voltmeter. Interfacing the digital voltmeter through the RDC into a time sharing system is one solution. Using the 5100, however, ensures continuous, uninterrupted service without the need for a long connect time to a large computer.

The outputs from the digital voltmeter and the control unit of the magnetometer are digital signals with magnitudes ranging from 0 to 5 volts. These signal levels are directly compatible with the input levels of the RDC's digital input module. This module also has a digital output point which transmits a pulse to step the analog signal multiplexer whenever the module is commanded to transfer data.

operation of the magnetometer experiment

The functional operation of this experiment is as follows: An APL program residing in the 5100 operates the SIO adapter by means of shared variables. The SIO adapter permits the program to communicate with the RDC and its attached equipment (see Figure 3). The message issued to the RDC is IB 3RB1, which causes the digital input module with address B to load the number 3 into its internal register, and then, using function 1, to read the data at its input connector.

Figure 3 Magnetometer experiment using IBM 5100 and RDC



The action of reading and the execution of the function are not performed until the system communication module detects that the 5100 is prepared to accept data. At that instant, the digital input module transmits a pulse to the multiplexer, causing it to step to the first position. The module then places itself in a WAIT state. Upon completion of integration, the digital voltmeter generates a DATA READY signal which satisfies the WAIT condition. The input data are then transmitted serially to the 5100.

The process is repeated until all three parameters have been read and the digital input module's preset count is exhausted. At that time, the system communication module generates an end-of-text character, which informs the 5100 that transmission is complete. This simple operation is repeated a number of times at each temperature setting. At the conclusion of a run, the user's program analyzes the data, and, via the RDC's analog output module, generates a graphic display on an X-Y plotter.

In applications of this nature, the digital output, process interrupt, and digital input processing are all performed by the digital input module in the RDC. Thus the 5100 is relieved of these tasks, and application programming is greatly simplified for the user. After the data have been captured, analyzed, and stored on the 5100 tape, the 5100 can be put in terminal mode, and the data and results passed on to a larger system for further analysis and storage in a data base.

**photoemission
electron
energy
distribution**

The second example application is an investigation of photoemission electron energy distribution. An important tool for probing the interactions between atoms is analysis of the energy distribution of electrons ejected from the material by the absorption of light. Much information can be obtained by using soft X rays to eject the electrons, and an excellent tunable source of soft X rays is synchrotron radiation. The number of synchrotrons is limited, however, and the experimentalist must travel to them to set up his experiment and acquire his data.

The key element in this experiment is capturing the energy spectrum. Typically this is done by using a multichannel analyzer, itself a small, special-purpose computer. Moving the data from the analyzer to a general-purpose computer for data analysis can be accomplished with the RDC. When the RDC is used in conjunction with the 5100, the whole apparatus can be transported readily to available sources of radiation.

**operation
of the
photoemission
experiment**

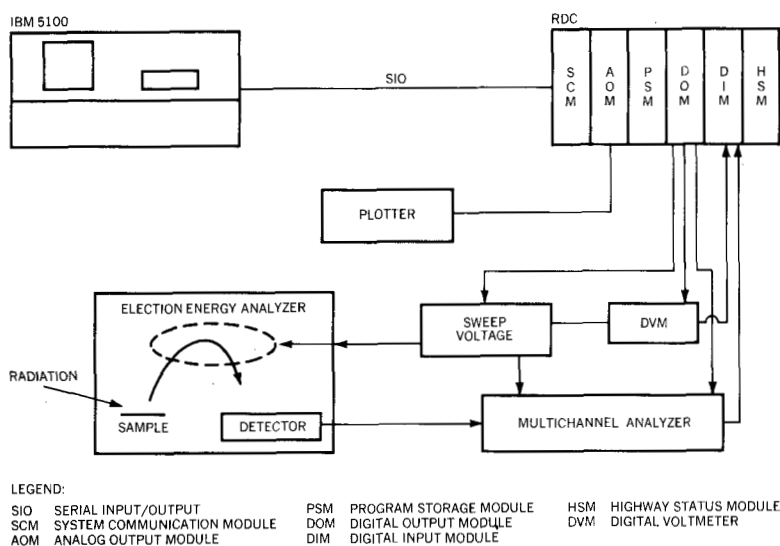
As with the magnetometer experiment, the user's program in the 5100 establishes shared-variable links to the SIO adapter, then issues messages to the digital output module which are used to select the desired sweep voltage range and speed (see Figure 4). These parameters are BCD digits which are passed via the digital output module to the sweep controller. The program verifies the starting and ending voltage values by reading a digital voltmeter attached to the digital input module. The data taking cycle is then started by issuing the necessary control signals, via the digital output module, to the multichannel analyzer controls.

After several hundred sweeps, the averaged data accumulated in the analyzer's memory are dumped into the 5100 via the analyzer's digital output and the RDC's digital input module. A new data run is then started. Typically each run requires 20 minutes or more. The use of the multichannel analyzer frees the 5100 and the RDC so that data from the previous run can be analyzed concurrently and displayed on an X-Y plotter attached to the analog output module. Again, the accumulated data and preliminary analyses, stored on 5100 tapes, can be passed on to a larger computer, through the terminal communication feature, for large-scale calculations and report preparation.

System limitations

The RDC is capable of transmission speeds, over start/stop lines, of up to 50000 baud. It can hold any mix of analog and digital input and output up to the limits imposed by available module positions. Thus the RDC can provide up to 448 digital input points or 148 digital output points, or 56 analog input points or 14 analog output points, or a mixture of the above.

Figure 4 Photoemission experiment using IBM 5100 and RDC



The maximum input transfer rate is determined solely by the speed of the link between the 5100 and the RDC. This link, the SIO adapter, is capable of speeds up to 9600 baud. Assuming that eight BCD or hexadecimal digits are transferred per measurement, at 9600 baud more than 125 measurements can be made per second. With analog input (two characters per ten-bit conversion), more than 500 measurements can be made per second.

For experiments in which the next step is directly related to an immediately preceding datum, the SIO auxiliary processor imposes an overhead of a third of a second for line turnaround. This responsiveness is completely predictable, unlike that of a time shared system, whose line turnaround responsiveness varies somewhat according to load.

In both of the experiments discussed above, all data transferred through the RDC are digital, so the reproducibility of the experiments is related only to the characteristics of the instrumentation itself. The analog input module could have been used instead of the digital input module in either experiment. The magnetometer experiment already had a satisfactory digital voltmeter, however, and cabling it to the digital input module was less expensive than using the analog input module. And in the photoemission experiment, the conversion and storage speeds required during a sweep were faster than the speeds that could have been achieved at 9600 baud, so the multichannel analyzer was the practical solution.

Summary

The IBM 5100 Portable Computer, used in conjunction with the Research Device Coupler, provides a personal laboratory automation system. Programming is greatly simplified by the availability of a single high-level language, either APL or BASIC, to perform not only numerical analysis, but also data acquisition and control. The RDC, with its simple command structure, provides for a variety of input/output functions directly at the experiment interface.

The two experiments described in this paper, in which high reliability and availability are primary requirements, are intended to exemplify the kinds of applications to which the 5100/RDC system can most advantageously be applied.

ACKNOWLEDGMENTS

The authors express their thanks to L. Kreighbaum, designer of the Research Device Coupler, and to S. Krasney, M. Lilie, and R. Dubke for their contributions and assistance. In particular, the authors thank H. Lilienthal, D. Eastman, and J. Freeouf, the experimenters, for their assistance with the descriptions of their experiments.

Appendix—5100/RDC programming formats

Following is a brief review of input and output programming formats for the IBM 5100 when operating in a stand-alone mode. APL is used in the examples. Output messages can be displayed on the 5100 screen by enclosing an alphanumeric character string in single quotation marks, as follows:

```
[7] 'ADJUST PWR SUPPLY VERNIER'
```

The message is displayed when statement 7 is executed. If the message is defined as a variable, the variable name in a statement causes the same execution:

```
[7] A←'ADJUST PWR SUPPLY VERNIER'  
[8] A
```

In this case, the message is displayed when statement 8 is executed.

Numeric data can be displayed in a similar manner:

```
[4] A←3 × 11.1  
[5] A
```

When statement 5 is executed, the product 33.3 is displayed.

Messages for the RDC must be passed to the 5100's auxiliary processor for transmission via the SIO adapter. The first step in this procedure is an offer to share a variable between APL and the auxiliary processor. For example:

```
[1] 1 □SVO 'DCOUT'
```

The digit 1 above refers to auxiliary processor number 1. The □SVO function offers the shared variable designated *DCOUT*. Acceptance or rejection of this offer is signified by a return code. A second global variable, *DCIN*, could be offered with statement 1, as follows:

```
[1] 1 □SVO 2 5 ρ 'DCOUTDCIN '
```

or

```
[1] 1 □SVO 'DCOUT'  
[2] 1 □SVO 'DCIN'
```

It is now necessary to select the command device option that informs the auxiliary processor about the nature of the SIO connection. Statement 3 selects the command subaddress 31.

```
[3] DCOUT←'OUT 31001 TYPE=I'
```

OUT indicates that information is to be passed to the command device. *TYPE=I* indicates that the data are in the form of a character string. Next, statement 4 selects the terminal option and a speed of 300 baud:

```
[4] DCOUT←'K/S,R/300'
```

Statement 5 closes the command device mode:

```
[5] DCOUT←10
```

At this point, no communication exists between the RDC and the 5100. Therefore the SIO path must be established. This is accomplished by issuing the following command to subdevice address 32:

```
[6] DCOUT←'OUT 32001 TYPE=I'
```

Statement 6 opens the SIO adapter for output data, either character or character string. The next statement issues an RDC message to the digital input module:

[7] $DCOUT \leftarrow CR, BS, 'IB\ 3RB'$

The *CR* and *BS* variables represent carriage return and back-space characters, respectively. Upon receiving these characters, the RDC accepts the balance of the message as commands. The RDC message in statement 7 is not transmitted until the output mode is closed. Statement 8 closes the output mode and forces the message to be transmitted:

[8] $DCOUT \leftarrow \text{10}$

The RDC message (statement 7) requests digital input module address B to load its preset counter with the number 3, then immediately read its input data three times. Reading will not take place until the system communication module detects that the 5100 is prepared to receive data. This is accomplished by opening the SIO adapter for input, as follows:

[9] $DCIN \leftarrow 'IN\ 33001'$

The first data reference via the SIO adapter, in statement 10, merely signifies that a link has been established:

[10] $Z \leftarrow DCIN$

Succeeding references will get input from the RDC. For example:

[11] $Z \leftarrow \overline{1} \downarrow Z \leftarrow 1 \downarrow DCIN$

Statement 11 operates from right to left. An end-of-transmission character is transmitted to the RDC, which alerts the system communication module that the 5100 has terminated transmission and is prepared to receive data. The system communication module then sends a beginning-of-text character and requests the digital input module to serialize, by character, the data at its input connector. The digital input module repeats the transmission of each field (that is, each measurement) three times, then informs the system communication module that transmission is complete. The system communication module then generates an end-of-transmission character.

Upon receiving the end-of-transmission character, the 5100 passes the data from the input buffer to the user's program. Statement 11 drops the beginning-of-text character from the buffer, then drops the end-of-text character. Thus the variable *Z* contains only the data. These steps can be written as an APL function for execution whenever data transfer is requested.

CITED REFERENCES

1. H. Cole, "System/7 in a hierarchical laboratory automation system," *IBM Systems Journal* **13**, No. 4, 307-324 (1974).
2. *IBM 5100 APL Reference Manual*, order number SA21-9213, IBM Corporation, Department 245, Rochester, Minnesota 55901 (1976).
3. *IBM 5100 APL Introduction*, order number SA21-9212, IBM Corporation, Department 245, Rochester, Minnesota 55901 (1975).
4. *IBM 5100 BASIC Introduction*, order number SA21-9216, IBM Corporation, Department 245, Rochester, Minnesota 55901 (1975).
5. *IBM 5100 BASIC Reference Manual*, order number SA21-9217, IBM Corporation, Department 245, Rochester, Minnesota 55901 (1976).
6. *IBM 5100 Communications Adapter Feature User's Manual*, order number SA21-9215, IBM Corporation, Department 245, Rochester, Minnesota 55901 (1976).
7. *IBM 5100 Serial I/O Adapter User's Manual*, order number SA21-9239, IBM Corporation, Department 245, Rochester, Minnesota 55901 (1976).
8. A. A. Guido, H. Cole, and L. B. Kreighbaum, "Interactive laboratory automation aid: The Research Device Coupler," *Proceedings of the IEEE* **63**, No. 10, 1509-1513 (1975).
9. H. Cole and A. A. Guido, *Device Coupler Guide*, Research Report RC4208, IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598 (1973).
10. H. Cole and A. A. Guido, *Device Coupler Functional Description*, Research Report RC4209, IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598 (1973).

Reprint Order No. G321-5044.